

# Dokumentation Ideathon Green-Coding & nachhaltige IT-Infrastrukturen

## Kompetenzentwicklung für nachhaltige IT

Ideathon-Team der Hochschule Bremerhaven, Stand 10. Feb. 2026

Ziele und Hintergrund des Ideathon.....	1
Fachlicher Hintergrund Green Coding.....	1
Workshops des Ideathon.....	2
Dokumentation der Workshopinhalte.....	4
Ideathon Workshop 1 - Green Coding in nachhaltigen Infrastrukturen.....	4
Ausblick.....	4
Material.....	4
Ideathon Workshop 2 - Messen des Energiebedarfs von Software.....	5
Diskussionsergebnisse:.....	6
Ausblick.....	7
Material.....	7
Ideathon Workshop 3 - Verbrauchsbesusste Sovereign Cloud Stack konforme Cloud Umgebungen.....	8
Wichtige Erkenntnisse des Workshops:.....	9
Ausblick.....	9
Material.....	9
Ideathon Workshop 4 - Green-Coding-Kompetenzen in Informatik- und IT-Lehrplänen.....	10
Zentrale Diskussionsergebnisse:.....	10
Ausblick.....	11
Material.....	11
Referenzen.....	12

## Ziele und Hintergrund des Ideathon

Am 9. Januar 2026 haben Vertreter\*innen des Informatik-Studiengangs der Hochschule Bremerhaven in Kooperation mit dem Bundesministerium Umwelt, Klimaschutz, Naturschutz und nukleare Sicherheit (BMUKN) und unter Mitwirkung von Vertreter\*innen des ECO:DIGIT-Projektes einen Ideathon durchgeführt, bei dem es darum ging, wie durch Green Coding und offene Softwarepraktiken nachhaltige digitale Anwendungen und Cloud-Infrastrukturen entstehen können. Zudem wurden Ideen gesammelt und konkretisiert, wie Green Coding und Kompetenzen für nachhaltige Softwareentwicklung stärkere Verbreitung in der Software-Ausbildung und -Praxis finden können.

Bei diesem Ideathon standen die kreative Ideenfindung, die schnelle Konkretisierung und die praktische Umsetzung im Vordergrund. In vier Workshops wurden innovative und nachhaltige Ansätze zum Thema "Green-Coding & nachhaltige IT-Infrastrukturen" erarbeitet.

Insbesondere für das praktische Coding wurde eine ressourcensparsame open-source-basierte containerisierte Infrastruktur des Studienbereichs Informatik der Hochschule Bremerhaven eingesetzt (siehe [Infrastruktur-Umgebung](#)). Der Zugang kann schnell am jeweiligen Veranstaltungsort eingerichtet werden und ist gleichzeitig ein Implementierungsbeispiel für eine nachhaltige Infrastruktur.

Kurzfristig wurde die Veranstaltung aufgrund der Witterungsverhältnisse auf ein hybrides Format umgestellt. Dazu hat das Ideathon-Team der Hochschule Bremerhaven ein einfaches Setting mit Open-Source-Werkzeugen (BigBlueButton, Hedgedoc, Excalidraw) eingerichtet, sodass Personen sowohl in Präsenz als auch online teilnehmen konnten.

## Fachlicher Hintergrund Green Coding

Green Coding bezieht sich auf die Entwicklung von Software mit möglichst geringem Energie- und Ressourcenverbrauch. In einer Workshopreihe zu Green Coding des Projektes Community Nachhaltige Digitalisierung des BMUKN wurden Ende 2023 Ansätze erarbeitet, um die Nachhaltigkeitsbemühungen in der Softwareentwicklung zu stärken. Die Ergebnisse wurden im [Booklet Green Coding](#) veröffentlicht. Dort wird Green Coding zum einen mit Bezug zu einem möglichst geringen Energieverbrauch der Software definiert. Zum anderen ist das Ziel von Green Coding bezogen auf die Hardwareressourcen "die Ermöglichung einer langen Nutzung der technischen Infrastruktur durch effizientere Software." (Green Coding Booklet, S. 5)

Konkrete Anforderungen und Kriterien für nachhaltige Software finden sich im Umweltzeichen [Blauer Engel für ressourcen- und energieeffiziente Software](#), mit dem Software zertifiziert werden kann, die entsprechende Nachhaltigkeits-Anforderungen erfüllt.

Für Entwickler\*innen bieten die Anforderungen dieses Umweltzeichens eine Leitlinie zur Identifizierung von Umweltbelastungen durch Software und für Ansatzpunkte, um diese zu reduzieren.

Verbraucher\*innen und staatliche Beschaffungsstellen können sich auf Basis der Zertifizierungen mit dem Blauen Engel für den Einsatz von Software entscheiden, die sparsam mit Hardwareressourcen umgeht und energiesparend ist.

Wie eine [Studie](#) der Gesellschaft für Informatik (GI) gemeinsam mit dem Umweltcampus Birkenfeld der Hochschule Trier und der Berliner Hochschule für Technik (htw) zeigt, werden Green Coding Kompetenzen jedoch bisher nur in wenigen Unternehmen umgesetzt, und Nachhaltigkeitskriterien spielen in der Softwareentwicklung oft noch keine angemessene Rolle. Dies liege vor allem an fehlender Vermittlung von Green-Coding-Kompetenzen:

„A lack of educational opportunities on Green Coding and the consequential lack of know-how prevents its effective implementation, as there are not enough software engineers with the specific knowledge and skills needed, and educational programs are not yet standardized through curricula.“ ([GI-Studie Potentials of Green Coding](#), S. 16)

Es zeigt sich also, dass es sowohl weiteren Forschungsbedarf im Hinblick auf Konzepte ressourcensparsamer Software- und Infrastruktur-Entwicklung gibt, als auch die Notwendigkeit, Green-Coding-Kompetenzen in IT-Ausbildungen zu verankern. In den Ideathon-Workshops wurden entsprechende Ansätze erarbeitet und diskutiert.

## Workshops des Ideathon

- Workshop 1 - Green Coding in nachhaltigen Infrastrukturen  
Mit Prof. Dr. Oliver Radfelder, Hochschule Bremerhaven zur Fragestellung: Wie können verteilte Softwarearchitekturen durch Entkopplung zum Beispiel mit redis beziehungsweise valkey nachhaltiger umgesetzt werden?
- Workshop 2 - Messen des Energiebedarfs von Software  
Mit Jan Ole Seutter, Franjo Gießel, Lamis Aiche und Lennart Steffen, Informatikstudierende der Hochschule Bremerhaven zur Fragestellung: Wie können wir mit einem containerisierten Setup eine hardware-unabhängige Messumgebung aufbauen, um den Energiebedarf von Software zu vergleichen?
- Workshop 3 - Verbrauchsbewusste Sovereign Cloud Stack konforme Cloud Umgebungen  
Mit Josefine Kipke und Felix Kronlage-Dammers, Open Source Business Alliance und ECO:DIGIT-Projekt zur Fragestellung: Wie können mit offenen Standards und Open Source Software nachhaltige Cloud-Lösungen realisiert und vergleichbar gemacht werden?
- Workshop 4 - Green-Coding-Kompetenzen in Informatik- und IT-Lehrplänen  
Mit Prof. Dr. Karin Vosseberg und Prof. Dr. Ulrike Erb, Hochschule Bremerhaven zur Fragestellung: Wie können Green-Coding-Kompetenzen als Querschnittsthemen und in praxis-/projektorientierten Modulen in IT-Curricula verankert werden: in Hochschule, Fachinformatikausbildung, Weiterbildung ...?

Die in der ersten Workshophälfte entwickelten Ideen wurden in der Mittagspause zwischen den Workshops ausgetauscht und anschließend weiterentwickelt mit dem Ziel praktischer Handlungsempfehlungen.

Während der Workshops wurden Ideen jeweils in einem gemeinsam bearbeitbaren Dokument (Hedgedoc) gesammelt. Die Ergebnisse der Workshops werden im Folgenden zusammenfassend dokumentiert.

Wir bedanken uns bei allen Teilnehmenden für die spannenden Diskussionen.

[Ulrike Erb](#)

[Karin Vosseberg](#)

[Oliver Radfelder](#)

für das Ideathon-Team der Hochschule Bremerhaven



CC BY NC SA: Karin Vosseberg

<https://informatik.hs-bremerhaven.de/ideathon/ideathon-uebersicht.pdf>

# Dokumentation der Workshopinhalte

## Ideathon Workshop 1 - Green Coding in nachhaltigen Infrastrukturen

Moderiert von Prof. Dr. Oliver Radfelder, Hochschule Bremerhaven

Ziel des Workshops war es, in einer konkreten Architektur, wie sie heutzutage nicht untypisch ist, Ansatzpunkte zu finden, um Green Coding und Nachhaltigkeit belegbar umzusetzen.

Zunächst wurde eine typische technische Architektur vorgestellt und eine konkrete Implementierung in der Open-Source-Infrastruktur der Hochschule Bremerhaven zum gemeinsamen Arbeiten zur Verfügung gestellt. Die technische Architektur besteht aus den bekannten Bausteinen **HAProxy** als Loadbalancer, **Tomcat** als Applicationserver, **PostgreSQL** als Datenbankserver und **Redis** als Key-Value-Store, Cache, Warteschlangenserver und Pub/Sub-Mechanismus. Mit diesen oder sehr ähnlichen, technischen Komponenten werden heute skalierbare Systeme aufgebaut: Der Loadbalancer sorgt dafür, dass Anfragen auf mehrere Applicationserver-Instanzen nach unterschiedlichen Verfahren je nach Last verteilt werden. Mit **Redis** wird das dann notwendige verteilte Session-Management betrieben, Caching umgesetzt und Warteschlangen für die asynchrone Ereignisverarbeitung eingesetzt.

Vorbereitet waren unterschiedliche Implementierungen von Ereignisweitergaben in verteilten Systemen, mit dem Ziel, ein besseres Verständnis der Effizienz von Producer-Consumer-Mustern, Ereigniswarteschlangen und Pub/Sub-Mechanismen mit unterschiedlichen Strategien in verteilten Systemen zu untersuchen.

Insbesondere wird **Redis** oder **ValKey** zu solchen Zwecken von Beginn an heute oftmals eingesetzt, selbst wenn die Art der Kommunikation einen hohen Ressourcenbedarf bedeutet. Die tatsächliche Möglichkeit den Bedarf in Relation zu anderen Architekturen zu messen, geht zudem in komplexen technischen Umgebungen, wie sie im Cloud-Computing üblich sind, oftmals verloren.

Aufgrund der Gegebenheiten vor Ort war es schwierig, den Workshop wie geplant umzusetzen. Die kurzfristig aufgebaute hybride Umgebung war dann doch wenig geeignet, tatsächlich gemeinsam zu coden. Zudem war die kleine Gruppe sehr heterogen in Bezug auf die alltägliche Coding-Praxis.

Es ergab sich aber schnell eine sehr grundsätzliche Diskussion über das Verhältnis von *klassischen Architekturen* aus dem Umfeld *Java Enterprise* zu solchen, die in den vergangenen Jahren aus der Open-Source-Bewegung hin zu asynchronem I/O mit *Javascript/node* als führender Vertreterin hervorgegangen sind.

Einig waren sich mehrere Teilnehmer:innen, dass sich der moderne asynchrone Ansatz zunächst einmal sehr viel leichtgewichtiger "anfühlt" und damit die Vermutung im Raum stand, dass mit der gefühlten Leichtgewichtigkeit auch eine Ressourcensparsamkeit einher geht. Die Sprache in der *Full-Stack-Szene* legt das gewiss auch nahe. Es gibt jedoch berechtigte Zweifel daran, dass diese Vorannahme auch belegbar ist.

Worauf sich geeinigt werden konnte, war, dass es eines erheblichen Aufwands bedürfte, um tatsächlich die unterschiedlichen Architekturen für konkrete Anwendungen vergleichen zu können. Es bedürfte mehrerer Szenarien, die in unterschiedlichen Architekturstilen umgesetzt werden, um überhaupt sinnvolle Messungen konzipieren und durchführen zu können. Letztlich bleibt es vorerst viel Erfahrungswissen, das an nachfolgende Generationen von Entwickler:innen weitergegeben wird.

Ein weiterer, ausführlich diskutierter Punkt war, dass Menschen in der Praxis bisweilen den Eindruck haben, umfängliche Regeln bezüglich Netzwerk trennung verhinderten all zu häufig die Anwendung guter, ressourcenschonender Praxis im Coding selbst. In der Zukunft sollte dieser Punkt gesondert untersucht werden.

## Ausblick

Im Hinblick auf die Entwicklung nachhaltiger Softwarearchitekturen bedarf es weiterer Untersuchungen dazu, welche Rolle konkret Redis und Valkey als Instrumente der losen Kopplung spielen können, und ob diese Art der Entkopplung zum Einsparen von Ressourcen beitragen kann.

## Material

Handout: <https://informatik.hs-bremerhaven.de/orahttps://informatik.hs-bremerhaven.de/ideathon/Ideathon-Workshops-Doku.pdf>

Systemumgebung für den 9.1.2026: <https://informatik.hs-bremerhaven.de/ideathon/umgebung-infra.html>

Report Green Coding - Deutsche Energie Agentur:

[https://www.dena.de/fileadmin/dena/Publikationen/PDFs/2024/REPORT\\_Green\\_Coding.pdf](https://www.dena.de/fileadmin/dena/Publikationen/PDFs/2024/REPORT_Green_Coding.pdf)

Videos zum Einstiegen:

- ByteByteGo: Top 5 Redis Use Cases <https://www.youtube.com/watch?v=a4yX7RUgTxI>
- Web Dev Simplified: Redis Crash Course <https://www.youtube.com/watch?v=jgpVdJB2sKQ>

## Ideathon Workshop 2 - Messen des Energiebedarfs von Software

Moderiert von Jan Ole Seutter, Lamis Aiche, Lennart Steffen und Franjo Gießel, Informatikstudierende der Hochschule Bremerhaven

In diesem sehr praktischen Workshop wurde untersucht, wie mit einem containerisierten Setup eine Hardware-unabhängige Messumgebung aufgebaut werden kann, um den Energiebedarf von Software zunächst zu messen und in einem weiteren Schritt vergleichbar zu machen. Unterschiedliche Ansätze aus verschiedenen Projekten wurden vorgestellt und der eigene Ansatz live vorgeführt. Darauf basierend wurden für einige Anwendungsszenarien Mess-Strategien entwickelt.

Ausgehend von einem anhaltend stark wachsenden Energiebedarf bei IKT-Infrastrukturen sowie dem in den Informationstechnologien nicht zu unterschätzenden Rebound-Effekt, bedarf es eines präzisen Monitorings des Energiebedarfs kleiner und großer Softwaresysteme, um fundierte Entscheidungen für Entwicklung und Nutzung nachhaltiger Systeme treffen zu können.

In den letzten zehn Jahren gab es viele Entwicklungen in diesem Bereich von unterschiedlichen Organisationen, Projekten und Forschungseinrichtungen. Ein anhaltendes Problem bleibt die fehlende Vergleichbarkeit von Ergebnissen unterschiedlicher Ansätze.

Im Workshop wurde das *Green Software Measurement Model* (GSMM) von Guldner et al. 2024 vorgestellt, in dem versucht wird durch ein standardisiertes Vorgehensmodell die Validität der Ergebnisse und ihre Vergleichbarkeit zu erhöhen. Außerdem wurden verschiedene Methodiken für unterschiedliche Fragestellungen vorgestellt. Mit den Teilnehmenden gab es einen Austausch über Herausforderungen und Strategien in der Praxis.

Grundsätzlich lassen sich die Mess-Methoden zur Ermittlung des Energiebedarfs von Software folgenden Ansätzen zuordnen:

- Externe Messgeräte
- On-Chip-Sensoren: Intel RAPL, NVIDIA NVML
- Reine Software-basierte Ansätze:  
Linux perf, PowerAPI
- Hochrechnungen & Modelle: SPECPower XGBoost Model

Je nach Fragestellung und Setup sind unterschiedliche Herangehensweisen gefragt. Die Qualität der Ergebnisse schwankt je nach Methode jedoch nach wie vor stark.

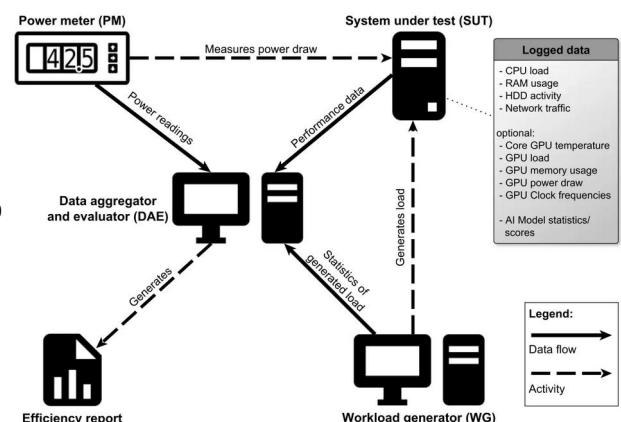
Bei unserem Ansatz, der [Projektgruppe KOSBUNEU](#), wird mithilfe eines leicht zu handhabenden, externen Messgerätes (einer Smart-Steckdose) die Leistungsaufnahme des Systems in Wh gemessen.

Mithilfe von Containerisierung (und wahlweise Virtualisierung) wird ein reproduzierbares Setup aufgebaut, in dem zunächst die Grundlast des Systems ermittelt wird und dann, in einem möglichst realistischen Nutzungsszenario, die zusätzlich benötigte Leistungsaufnahme des *System under Test* (SuT) dem zu messenden Software-System attribuiert wird.

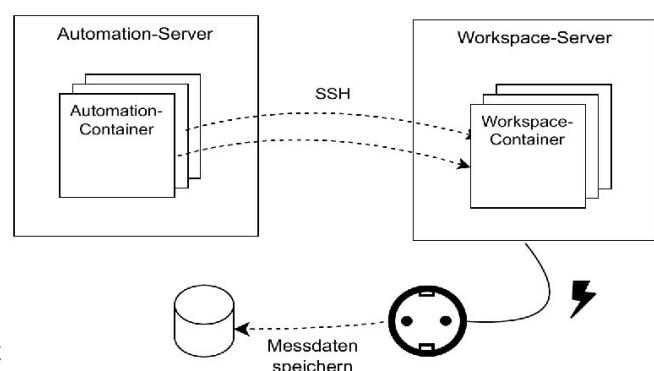
Vorteile des Setups sind: Es ist vergleichsweise leicht zu reproduzieren und für unterschiedliche Anwendungsszenarien anzupassen. Die Ermittlung der Messdaten sind über die REST-Schnittstelle der Smart-Steckdose leicht zu aggregieren und mit Tools wie *Grafana* oder *gnuplot* graphisch auswertbar. Die ermittelten Messdaten sind vergleichsweise präzise und von konstanter Qualität.

Nachteile sind: Der physische Zugriff auf die Stromversorgung muss gegeben sein. Die Vergleichbarkeit von Messdaten auf unterschiedlichen physischen Servern, insbesondere verschiedener Generationen ist nicht ohne Weiteres gegeben.

Im Workshop wurde das Setup vorgestellt, live vorgeführt und gemeinsam diskutiert.



Schematische Darstellung eines SERENA Messaufbaus  
(Quelle: Umwelt-Campus Birkenfeld (2025) <https://www.umwelt-campus.de/iss/projekte/laufende-projekte/kira/ergebnisse/>)



Architektur der Testinfrastruktur im Projekt [KOSBUNEU](#)

## Diskussionsergebnisse:

Für Teilnehmende interessante Messobjekte und -ziele

- Verteilte Anwendungen in einem Kubernetes-Cluster
- ERP-System Infrastruktur
- bestimme Usage- und Echtzeit-Messung im Live-Betrieb
- VMs messbar machen
- Haben Datenbank- oder Code-Optimierung wirklich einen Effekt?
- Stromverbrauch zu verschiedenen Web-Frameworks messbar machen
- Messung der Endgeräte und Energieverbrauch beim Client
- Wie kann man Ablösung von Altanwendung durch Neuanwendung messen. Wie unterscheidet sich der Energieverbrauch?

Weitere diskutierte Themenfelder:

- Methodologisch:
  - Sollte die ermittelte Grundlast für den ermittelten Energiebedarf eines Software-Systems miteinberechnet werden? (*statischer* Energiebedarf + *dynamischer* Energiebedarf)
  - Für beides gibt es Argumente: Wie gut lässt sich die Grundlast isolieren? Software benötigt i.d.R. ein OS; gesucht ist oft die Attribution der Leistungsaufnahme zu einem definierten Software-Stack bzw. Veränderungen des Energiebedarfs einer gegebenen Software (und nicht des darunter liegenden OSs).
  - Die Messung selber verbraucht auch Energie, fließt nicht in die Messergebnisse ein.
  - Für bestimmte Setups muss die Bildschirmzeit von Nutzenden mit einbezogen werden.
  - Netzwerkauslastung müsste mit gemessen werden.
  - Im Vergleich: Das *Green Metrics Tool* arbeitet bei jeder Messung mit den Phasen: Boot up, Idle, Auslastungszustand, Boot down.
  - Schwierigkeit: den Ressourcenverbrauch vom Messtool herauszufiltern.
  - Netzwerkverkehr kann mit 0,1 Watt pro Gigabit eingerechnet werden -> Ecograder JSO2 Mittelwerte Annäherung (zu überprüfen).
- Bezgl. Blogpost: "[Energiebedarf von Software: Eine Anleitung zum selber machen](#)": Wie valide ist die Methodik?
  - Formel: Energieverbrauch=CPULaufzeit/CPUKernel\*maximaleLeistungsaufnahme (Bsp.: 15,57s/4\*11W=42,8Ws).
  - Die Messung ist ungenau, solange das Energieprofil der CPU nicht einbezogen wird, - auf der anderen Seite ist dies ein grundsätzliches methodologisches Problem bei der Nutzung unterschiedlicher Hardware für das *System under Test* (SuT).
- Sollte auch für Endnutzer:innen sichtbar sein, wieviele Ressourcen einzelne Apps verbrauchen?
  - Awareness bei der Nutzung ist wichtig (vgl. *Reboundeffekt*).
  - Systeme sichtbar machen, um Einsparungen umsetzen zu können  
-> Auch für Unternehmen sehr relevant, um Kosten zu senken.
  - Es kann viel bringen, wenn Nutzer:innen mehr über ihre Nutzung wissen,
  - Gerade jetzt, mit steigender Nutzung von KI, sollte deutlicher kommuniziert werden, was Nutzer:innen mit ihren Endgeräten gerade verbrauchen.
  - Anfragen mit einem Verbrauch von 0,3 Wattstunden in der Google-Suchmaschine werden bei ChatGPT auf 2,9 Wattstunden geschätzt (zu überprüfen).
  - GPT5 verbraucht 8x soviel Energie wie GPT 4 (zu überprüfen).
  - Beispiel für ein Energieeffizienz-Label im Bereich Maschinelles Lernen:  
<https://lamarr-institute.org/de/blog/energieeffizienz-im-ml/>
- Weitere Tools:
  - Software Carbon Intensity (SCI) Specification ("[...] defines a methodology for calculating the rate of carbon emissions for a software system"): <https://sci.greensoftware.foundation/>

- Power Hog - Tool(s) zum Messen des Energieverbrauchs von Endgeräten (derzeit nur MacOS): <https://www.green-coding.io/de/products/power-hog/>
- CO2e-Fußabdruck von Websites messen und vergleichen: <https://ecograder.com/>
- PowerAPI “Measuring power consumption of your applications” Modell-/ KI-basierter Ansatz: <https://powerapi.org/>
- Alternative zu k6, um automatisiert durchzuführen:  
Puppeteer: <https://pptr.dev/guides/page-interactions>
- Zum Thema effizientes / nachhaltiges Softwareengineering / CI-Pipelines:  
<https://carbon-aware-sdk.greensoftware.foundation/>  
<https://www.green-coding.io/de/products/eco-ci/>
- CO2.js: “[...] JavaScript library that enables developers a way to estimate the emissions related to use of their apps, websites, and software”  
Repo: <https://github.com/thegreenwebfoundation/co2.js>  
Dokumentation: <https://developers.thegreenwebfoundation.org/co2js/overview/>
- Benchmark-Vergleich von Javascript Frameworks (Stefan Krause)  
Ergebnisse:  
[https://krausest.github.io/js-framework-benchmark/2025/table\\_chrome\\_143.0.7499.41.html](https://krausest.github.io/js-framework-benchmark/2025/table_chrome_143.0.7499.41.html)  
Repo: <https://github.com/krausest/js-framework-benchmark>
- Full environmental footprint of AI *Cradle-to-Grave environmental impacts of GenAI training on the Nvidia A100 GPU* (Falk et al. 2025) - Messbarkeit von Nvidia GPUs für KI-Training: <https://arxiv.org/abs/2509.00093>  
Dazu die Präsentation:  
<https://drive.google.com/file/d/1ZjHXRBHQxssBJZHwXI1bLdojhN8dHn/view?usp=sharing>
- Übersicht über viele weitere Tools und Projekte, rund um das Thema Software Messen (Bundesverband Green Software):  
<https://landscape.bundesverband-green-software.de/?group=measurement>

## Ausblick

Die Diskussionsergebnisse und offenen Fragen zeigen, dass es in diesem Bereich noch großen Forschungsbedarf gibt. Wir werden uns insbesondere mit den folgenden Fragen weiter beschäftigen und suchen dazu den Austausch mit Interessierten:

- Wie kann man Messergebnisse derselben Software auf unterschiedlichen Hardware-Plattformen vergleichbar machen? (Vgl. *Software Carbon Intensity (SCI) Specification* s.o.)
- Wie könnten konzeptionell Software-Produkte untereinander vergleichbar gemacht werden, sodass sie z.B. in *Energieeffizienzklassen* eingeordnet werden können?

## Material

- Link zum Workshop-Repo: <https://gitlab.informatik.hs-bremerhaven.de/ideathon2/ideathon-workshop>
- Link zu den Workshop-Folien: [https://informatik.hs-bremerhaven.de/ideathon/WS2-Folien\\_compr.pdf](https://informatik.hs-bremerhaven.de/ideathon/WS2-Folien_compr.pdf)
- Für einen leichten Einstieg: Gröger, Jens (2022): Energieverbrauch von Software: Eine Anleitung zum selber machen. Blogpost Öko-Institut. <https://www.eko.de/blog/energieverbrauch-von-software-eine-anleitung-zum-selbermessen/>
- Detaillierter Überblick über verschiedene Projekte und Ansätze: Guldner, Achim et al. (2024): Development and evaluation of a reference measurement model for assessing the resource and energy efficiency of software products and components—Green Software Measurement Model (GSMM), Future Generation Computer Systems, Volume 155, 2024, pp. 402-418.  
<https://www.sciencedirect.com/science/article/pii/S0167739X24000384>
- Institut für Softwaresysteme, Umwelt-Campus Birkenfeld (2025): KI-Referenzmodell für Energie- und Ressourceneffizienz und dessen industrielle Anwendung (KIRA).  
<https://www.umwelt-campus.de/iss/projekte/laufende-projekte/kira/ergebnisse>
- Weitere Referenzen siehe [Folien](#)

## Ideathon Workshop 3 - Verbrauchsbesusste Sovereign Cloud Stack konforme Cloud Umgebungen

Moderiert von Josefine Kipke und Felix Kronlage-Dammers, OSBA und ECO:DIGIT-Projekt

Dieser Workshop befasste sich mit der Frage: Wie können mit offenen Standards und Open-Source-Software nachhaltige Cloud-Lösungen realisiert und vergleichbar gemacht werden?

Ausgangspunkt waren bisherige Ergebnisse des ECO:DIGIT-Projektes, das den ökologischen Fußabdruck von Software und digitaler Infrastruktur über deren gesamten Lebenszyklus hinweg bewertet. Um einen Eindruck davon zu gewinnen, wie unterschiedlich das Verständnis der Teilnehmenden zu Begriffen wie „Cloud Computing“ und „digitale Souveränität“ sein kann, haben wir einige interaktive Fragen gestellt.

### Was bedeutet „digitale Souveränität“ für deine Organisation?



Die Themengebiete rund um SCS und ECO:DIGIT sowie das Nachhaltigkeitsmonitoring in Cloud-Infrastruktur sind komplex. Um eine gemeinsame Gesprächsgrundlage zu schaffen, gab es zunächst zwei Impulsvorträge. Dabei ging es um SCS und Leaf, letzteres ist eine Monitoring-Lösung für SCS-konforme Clouds, welche im Rahmen von ECO:DIGIT entwickelt wurde.

In mehreren Arbeitsgruppen wurde anschließend, über die Anforderungen an einen Score diskutiert, mit dem sich verschiedene Cloud-Umgebungen anhand ihres Ressourcenverbrauchs vergleichen lassen. Da einige Besucher aufgrund des starken Schneefalls nicht anreisen konnten, fand ein Großteil der Teilnehmenden online zusammen. Um eine stabile Kollaboration zu ermöglichen, wurde ein Excali-Whiteboard bereitgestellt, auf dem sich beispielsweise einige Hilfsmittel sowie Leitfragen für die Erarbeitung eines solchen Scores finden ließen. Interessant war, dass sich die Gruppen jeweils unabgesprochen ganz unterschiedlichen Schwerpunkten widmeten. Dadurch ergaben sich beim Vorstellen und Besprechen der Ergebnisse zahlreiche Gespräche.

Die Ergebnisse des Workshops fließen in den zu entwickelnden Score ein. Dieser wird anschließend im Rahmen des ECO:DIGIT-Projekts in das Nachhaltigkeitsmonitoring für SCS-konforme Cloud-Umgebungen umgesetzt.

Ziel: Erarbeitet die Grundlagen für einen standardisierten Vergleichscore von Rechenzentren SCS konformer Cloud Umgebungen basierend auf Umweltwirkungen und Verbrauchsmetriken.

#### Leitfragen für die Recherche

##### 1. Vergleichbarkeit absoluter Verbrauchswerte

- Wie kann man absolute Verbrauchswerte (Energieverbrauch, Wasser, CO<sub>2</sub>e) zwischen unterschiedlichen Cloud-Anbietern vergleichbar machen?
- Welche Normalisierungsmethoden existieren? (z.B. pro Workload, pro Betriebsstunde, pro Rechneroperation)
- Wie wird die Heterogenität von Rechenzentren (Größe, Nutzungsgrad, Geografischer Standort) berücksichtigt?
- Welche Datenquellen und Standards sind verlässlich?

##### 2. Bestehende Ansätze und Methoden

- Gibt es bereits Ansätze oder Methoden, um Vergleichs-Scores zu bilden?

##### 3. Umweltwirkungen und Verbrauchsmetriken

- Welche Umweltwirkungen oder Verbrauchsmetriken sollten einfließen?
- Energie: Welche Aspekte? (Gesamtverbrauch, Energiemix/Kohlenstoffintensität, Erneuerbare?)

#### Erarbeitungsprozess

##### Recherche durchführen

- Nutzt Fachliteratur, Websuche, Standards (GRI, Greenhouse Gas Protocol, Perplexity)

##### Findings sammeln

- Notiert Methoden, Standards, Lücken auf dem Whiteboard

##### Vergleich erstellen

- Tabellarische Übersicht: Ansatz / Metriken / Normalisierung / Bewertung

##### Kritische Reflexion

- Wo sind Probleme?

##### Anforderungen für euren Score

- Was braucht ihr mindestens, was ist optional?

## Wichtige Erkenntnisse des Workshops:

- Vergleich des Verbrauchs: Wie können wir Energie-, Wasser- und CO<sub>2</sub>-Kennzahlen über Anbieter hinweg standardisieren? Als Lösungen wurden Standard-Workloads und kontinuierliche Berichterstattung vorgeschlagen.
- Vielfalt der Rechenzentren: Größe, Auslastung und Standort beeinflussen die Effizienz: Eine höhere Auslastung reduziert den Kühlungsaufwand, während die geografische Lage die Wassereffizienz beeinflusst.
- Zuverlässige Standards: Der Cloud-Verhaltenskodex, ISO 14001 (Umweltmanagement) und ISO 50001 (Energiemanagement) erweisen sich als wesentliche Rahmenwerke.
- Vergleichende Bewertung: Eine ganzheitliche Bewertung sollte betriebliche und inhärente Emissionen, Wasserverbrauch, Kreislaufwirtschaftspraktiken und Transparenz einbeziehen.

## Ausblick

Im Workshop wurde mit einigen der Student\*innen der HS Bremerhaven außerdem eine Breakout-Session zu SCS gehalten. Dabei wurde konkret über SCS gesprochen und der Umfang der Beschäftigung mit SCS erörtert. Ein weiterer Austausch in Bremerhaven ist in Planung.

## Material

- Whiteboard: [https://excalidraw.com/#room=7bfc1b78e51debbd48b1,UC\\_-XUxR6x8yQnCqAWFaTQ](https://excalidraw.com/#room=7bfc1b78e51debbd48b1,UC_-XUxR6x8yQnCqAWFaTQ)
- Folien: <https://informatik.hs-bremerhaven.de/ideathon/ideathon-workshop3-folien1.pdf>  
<https://informatik.hs-bremerhaven.de/ideathon/ideathon-workshop3-folien2.pdf>
- Sovereign Cloud Stack: <https://sovereigncloudstack.org/ueber-scs/>
- Virtualisierung: <https://www.ibm.com/de-de/think/topics/virtualization>
- Deep dive - Container vs VMs: <https://www.ibm.com/think/topics/containers-vs-vms>
- Cloud Platform OpenStack (IaaS): <https://www.openstack.org/>

# Ideathon Workshop 4 - Green-Coding-Kompetenzen in Informatik- und IT-Lehrplänen

Moderiert von Prof. Dr. Karin Vosseberg und Prof. Dr. Ulrike Erb, Hochschule Bremerhaven

Basierend auf den Themen der anderen drei Workshops ging es in Workshop 4 darum, wie Green-Coding-Kompetenzen sowohl als Querschnittsthemen in IT-Curricula verankert als auch in expliziten praxisorientierten Green-Coding-Modulen vermittelt werden können.

Ausgangspunkt der Diskussionen waren Untersuchungsergebnisse, wonach mangelnde Umsetzung von Green Coding in der Softwareentwicklung sowohl an mangelnden Kompetenzen als auch an mangelndem Problembewusstsein in den IT-Abteilungen vieler Unternehmen liegt (GI 2023: [Potentials of Green Coding: Perspectives on Implementation](#). Ergebnisse einer GI-Befragung). Zudem machten Teilnehmer\*innen aus Schleswig-Holstein deutlich, welche Bedeutung entsprechende Kompetenzen auch z.B. im staatlichen Beschaffungswesen haben, um fundierte Entscheidungen über die Beschaffung nachhaltiger Software treffen zu können.

Der Workshop profitierte von den vielfältigen Perspektiven der Teilnehmenden aus Hochschule und Weiterbildung, von Lehrenden, Studierenden, Qualitätssicherung und Digitalisierungsberatung.

In einer Brainstormingphase wurden Ideen gesammelt, welche Green-Coding-Aspekte in einem IT-Lehrplan enthalten sein sollten. Im Anschluss wurde anhand eines konkreten Informatik-Curriculums der Hochschule Bremerhaven diskutiert, in welchen Modulen dieses Curriculums bereits Green-Coding-Anteile vermittelt werden und wo dies noch vermehrt implementiert werden könnte.

## Zentrale Diskussionsergebnisse:

Green Coding – Querschnittsthema oder eigenes Modul?

- Insbesondere aus Sicht von Studierenden sollte Green Coding eher als Querschnittsthema im Curriculum verankert werden. Hier sollten stets verschiedene Aspekte der Nachhaltigkeit mitgedacht werden.
- Im Curriculum sollte sichtbar gemacht werden, wo und inwiefern Nachhaltigkeit als Querschnittsthema behandelt wird.
- Dabei ist zu bedenken, dass nicht nur Nachhaltigkeit sondern auch andere Querschnittsthemen wie Qualitätssicherung und Sicherheit zu integrieren sind.
- Vorschlag: Wissen vermitteln, wenn es gebraucht wird => Projektorientierung
- Eine explizite Green-Coding-Veranstaltung könnte zusätzlich als Wahlfach angeboten werden (als Vertiefungsmodul). Der exemplarische Entwurf für ein solches Wahlmodul findet sich unter <https://informatik.hs-bremerhaven.de/ideathon/CurriculumModulGreenCodingBeispiel.pdf>

Curriculum						
Zeit	Montag	Dienstag	Mittwoch	Donnerstag	Freitag	

Weitere wichtige Kompetenzen im Umfeld von Green Coding:

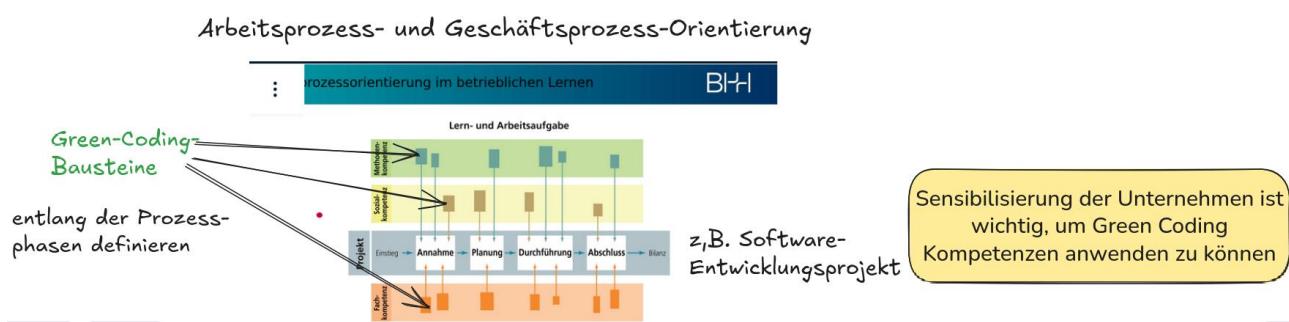
- In einem Wahlmodul Nachhaltige Softwareentwicklung sollten berücksichtigt werden: gesellschaftliche Perspektive, Auswirkungen, ethische Aspekte, Handlungsspielräume, Verantwortung von Softwareentwickler:innen
- Wissen über bereits vorhandene Green Coding Frameworks und nachhaltige Algorithmen vermitteln
- Problembewusstsein und Folgenabschätzung
- Wissen um Standards und Zertifikate (u.a. Blauer Engel)
- Monitoring / messbar machen des Ressourcenverbrauchs
- Wissen haptisch greifbar machen
- kommunikative Kompetenzen, um z.B. argumentieren zu können, warum Green Coding ein Vorteil für Unternehmen ist.
- Projekte versus Grundlagenwissen - ist das ein Gegensatz?

Auf dem Weg zur Etablierung von Green-Coding-Curricula:

- Barcamps, Summer Schools, Hackathons mit Fokus auf Nachhaltigkeit können der Sensibilisierung dienen.
- Bildungsakteure können einen Werkzeugkasten mit Green-Coding-Bausteinen zusammenstellen und zur Verfügung stellen mit kleinen Lerneinheiten und Best-Practice-Beispielen
- Green-Coding-Bausteine entlang der Arbeitsprozesse identifizieren siehe [Folien von Henning Klaffke](#)
- Handlungsorientierung
- Für die Integration von Green-Coding-Kompetenzen in Weiterbildungs-Lehrpläne könnten sie nach Handlungsebenen strukturiert werden, z.B.:

## Exemplarischer Ansatz zur Strukturierung von Green-Coding-Kompetenzen nach Handlungsebenen (unvollständiger Entwurf)

<b>Handlungsebenen zur Strukturierung von Green-Coding-Kompetenzen in der IT-Ausbildung</b>	<b>sachgerecht nachhaltiges Handeln</b>	<b>gesellschaftlich verantwortliches Handeln</b>
<b>berufsspezifisch-profilgebende Ebene</b>	Entwicklung ressourcensparsamer langlebiger Software	Entwicklung von Open Source Software, Unterstützung von Open-Source-Projekten und -Communities; public money – public code
<b>betrieblich-organisatorische Ebene</b>	Ressourcensparsame Prozesse gestalten	Ressourcensparsame Softwarenutzung für Endanwender ermöglichen
	IT-Beschaffung auf Basis von Umwelt-Gütesiegeln und Nachhaltigkeitskriterien	Nachhaltige Lieferketten beachten
	Einsatz nachhaltiger Anwendungssoftware (langlebig, lauffähig auch auf älterer Hardware, ohne Vendor-Lock-in, modular, anpassbar usw., siehe Kriterien des Blauen Engel für ressourceneffiziente Software)	
	Nutzung nachhaltiger souveräner IT-Infrastrukturen	Ökonomische digitale Unabhängigkeit erreichen
<b>gesellschaftliche-unternehmerische Ebene</b>	CO2-Fußabdruck / Energieverbrauch über den gesamten Lebenszyklus messen	
		Beachtung rechtlicher und gesellschaftlicher Rahmenbedingungen (Informationelle Selbstbestimmung, Datenschutz, Digitale Souveränität, Nachhaltigkeitsziele usw.)



## Ausblick

Wir werden die Idee weiterverfolgen, eine Art Werkzeugkasten zusammenzustellen mit Best-Practice-Bausteinen zur Integration von Green-Coding-Inhalten in Lehrveranstaltungen und Curricula. Dafür und für die weitere Entwicklung von Green-Coding-Curricula kooperieren wir gerne mit den interdisziplinären Teilnehmerinnen und Teilnehmern dieses Workshops.

## Material

- Intro und Impulse: <https://informatik.hs-bremerhaven.de/ideathon/ideathon-workshop4-intro.pdf>
- Entwurf für ein exemplarisches Green-Coding-Wahlmodul: <https://informatik.hs-bremerhaven.de/ideathon/CurriculumModulGreenCodingBeispiel.pdf>
- Folien von Henning Klaffke, BHH, zur Gestaltung arbeitsprozessorientierter Lernumgebungen: <https://informatik.hs-bremerhaven.de/ideathon/BHH-Klaffke-Arbeitsprozessorientierung.pdf>
- Brainstorming: <https://excalidraw.com/#room=1a2912ca0852c41eef79,S4VMrFiSIGhof-exCIRBFQ>

## Referenzen

- BMUKN (2024): Green Coding – Booklet zur Workshoepreihe. Community Nachhaltige Digitalisierung – Workshop Green Coding.  
[https://www.bmuv.de/fileadmin/Daten\\_BMU/Download\\_PDF/Digitalisierung/das\\_green\\_coding\\_booklet\\_bf.pdf](https://www.bmuv.de/fileadmin/Daten_BMU/Download_PDF/Digitalisierung/das_green_coding_booklet_bf.pdf)
- Deutsche Energie-Agentur (Hrsg.) (dena, 2024) „Green Coding – Mit stromsparender Software zu einer nachhaltigeren Digitalisierung“.  
[https://www.dena.de/fileadmin/dena/Publikationen/PDFs/2024/REPORT\\_Green\\_Coding.pdf](https://www.dena.de/fileadmin/dena/Publikationen/PDFs/2024/REPORT_Green_Coding.pdf)
- UBA (2020): Blauer Engel für ressourcen- und energieeffiziente Softwareprodukte, DE-UZ 215 - Vergabekriterien. Ausgabe Januar 2020, Version 4.  
<https://produktinfo.blauer-engel.de/uploads/criteriafile/de/171/DE-UZ 215-202001-de Kriterien-V4.pdf>
- Bretschneider, Markus & Jil Schirner (2025): Nachhaltigkeit in Ausbildungsordnungen – Umsetzungskonzept und Materialien im Überblick. Herausgegeben vom Bundesinstitut für Berufsbildung, Bonn.  
[https://www.bibb.de/dienst/dapro/daprodocs/verweise/so\\_22406\\_Überblick\\_Konzept\\_Materialien.pdf](https://www.bibb.de/dienst/dapro/daprodocs/verweise/so_22406_Überblick_Konzept_Materialien.pdf)
- GI (2023): Potentials of Green Coding: Perspectives on Implementation. Documentation. Ergebnisse einer GI-Befragung. Online: [https://gi.de/fileadmin/user\\_upload/ISOC/2023-09-25\\_Green\\_Coding\\_final.pdf](https://gi.de/fileadmin/user_upload/ISOC/2023-09-25_Green_Coding_final.pdf)
- Gröger, Jens (2022): [Energieverbrauch von Software: Eine Anleitung zum selber machen](#). Blogpost Öko-Institut. <https://www.oeko.de/blog/energieverbrauch-von-software-eine-anleitung-zum-selbermessen/>
- Guldner, Achim et al. (2024): Development and evaluation of a reference measurement model for assessing the resource and energy efficiency of software products and components—Green Software Measurement Model (GSMM), Future Generation Computer Systems, Volume 155, 2024, pp. 402-418. <https://www.sciencedirect.com/science/article/pii/S0167739X24000384>
- Junger, D.; M. Westing; C. P. Freitag; A. Guldner; K. Mittelbach; K. Obergöker; S. Weber; S. Naumann; V. Wohlgemuth (2024): Potentials of Green Coding - Findings and Recommendations for Industry, Education and Science. Extended Paper. arXiv. <http://arxiv.org/abs/2402.18227> In: INFORMATIK 2023 - Designing Futures: Zukünfte gestalten pp. 1289-1299  
(siehe auch <https://gi.de/en/aktuelles/projekte/en-green-coding>)